International Journal of Scientific Research & Engineering Trends
Volume 12, Issue 6, Nov-Dec-2026, ISSN (Online): 2395-566X

Adaptive A IoT Security Framework for Real-time Anomaly and Intrusion Detection in Connected Devices

Nafisa S[1], Dr. Balaji.K[2]
[1]Research Scholar
nafisasheriff@gmail.com)
[2]Research Supervisor
koturu@gmail.com)

Abstract- The emergence of AI and IoT (AIoT) technology fusion has opened up a new avenue for the creation of highly insecure conditions in terms of protecting continuous streams of information generated by the heterogeneous population of connected devices from highly advanced attacks. Existing static approaches suffer from limitations related to concept drift, limited resources and inherent dynamics of threat space. We introduce Adaptive AIoT Security Framework (AASF) combining lightweight edge-based anomaly detection with federated collaborative learning and online ensemble adaptation techniques. The proposed solution is built on the basis of two-level architecture: a resource-optimized sparse autoencoder used for computing local anomaly scores on resource-constrained devices, and federated aggregation layer hosted in the cloud that helps to keep model synchronization at regular intervals without revealing telemetry data. An adaptive thresholding technique relying on reconstruction error distribution allows an individual edge device to set up its sensitivity level for detection. Experimental results show that AASF outperforms existing solutions with respect to the detection performance while ensuring practical applicability on edge devices, as well as minimizing the amount of required communication overheads through selective updating.

Key Word: AIoT Security, Anomaly Detection, Intrusion Detection, Federated Learning, Edge Computing, Adaptive Thresholding, Concept Drift

I. INTRODUCTION

With the rise of IoT devices, there have been radical changes in the computing paradigm, where pervasive sensing and actuation have become possible in sectors from industrial automation to healthcare [1], [2]. Since these IoT devices come with the capability of artificial intelligence at the edge of the network, an AIoT system emerges as a highly autonomous and responsive system. But with the intelligence on the edge, there will be an enormous increase in the attack surface, whereby every sensor, actuator, and gateway on the edge becomes vulnerable to attacks [1], [2]. This security problem is also made more complicated with the inherent nature of the AIoT environment in which resource-constrained hardware, heterogeneous communication channels, intermittent connectivity, and physical availability of the edge devices make centralized security solutions ineffective [1], [10].

NIDS have always been an essential element in defensive infrastructure; however, their use in AIoT systems shows a number of shortcomings [1]. Signature-based NIDS cannot detect attacks with previously unseen attack patterns, while machine learning algorithms, which require static training sets, will degrade quickly due to changing configuration of devices, traffic dynamics, and attacks' techniques [1], [10]. In addition, the problem of concept drift—a change in the statistical properties of network traffic—is especially urgent in the context of AIoT, where new devices emerge and firmware updates, as well as other factors, continually alter the baseline of operation. In addition, the problem of extreme class imbalance in data on intrusion detection, where malicious actions take up only a tiny portion of all traffic, makes the training of classifiers even more difficult [7].

In recent times, federated learning has been studied as a promising privacy-preserving solution to implement distributed anomaly detection in the absence of centralized raw data [4], [9], [10]. Nevertheless, naive federated learning suffers from heavy communication costs and cannot tackle non-identically distributed datasets over diverse edge devices [4], [9]. Lightweight models based on knowledge distillation have shown success in edge environments, although the inability of fixed temperatures and student models to evolve

© 2026 IJSRET

according to changing traffic patterns has become a bottleneck for their success [6], [8]. Online ensembles provide an avenue to mitigate concept drift by virtue of continuous updating of the models, but still pose challenges in incorporating them together in one framework [7].

The proposed work bridges this gap with the introduction of the Adaptive AIoT Security Framework (AASF), which presents a comprehensive architectural approach toward real-time anomaly and intrusion detection in connected device ecosystems. The proposed framework has three main contributions. The first one involves the development of a lightweight sparse autoencoder architecture tailored to run on microcontroller-class devices, which can perform local anomaly scoring in real time in an energy-efficient way. The second contribution relates to the design of an adaptive thresholding algorithm that adapts the sensitivity of detection based on the reconstruction error distribution at each node, without any user input. Finally, it implements a federated aggregation protocol with selective update exchange, where the number of communication rounds required is decreased.

The rest of the paper is divided into the following sections. The Literature survey describes related literature in the domain of AIoT security, federated learning for intrusion detection, and design of light-weight models. The Proposed Methodology discusses the AASF framework in detail along with algorithms used and issues pertaining to implementation. The Analysis section contains the results of comparison analysis.

II. LITERATURE SURVEY

The field of AIoT security and intrusion detection has seen an increase in its literature, which indicates the significance and the technical variety associated with this field. Early surveys have identified the fundamental categories of IoT risks and countermeasures, in which Al-Garadi et al. gave a detailed survey of machine learning and deep learning for IoT security which is still widely referred to [1]. This literature emphasized the trade-off between detection precision and feasibility of computation on constrained devices [1]. Further studies in this field have advanced our knowledge, in which Obaidat et al. listed down the security and privacy issues in layered IoT architecture and the added risks posed by edge intelligence [2].

One emerging research direction that is gaining attention is that of developing lightweight architectures that can be deployed at the edge. LiteKD is a method introduced in 2026 where a hybrid CNN-GRU teacher architecture is condensed into a small student architecture through the use of knowledge distillation [6]. This method uses a new adaptive soft-labeling scheme that dynamically regulates the process of knowledge distillation depending on the temperature and thus overcomes the challenge of fixed temperature in previous methods [6]. While showing good detection accuracy on benchmark data sets, LiteKD relies on a fixed teacher architecture and lacks adaptation schemes for concept drift [6]. On the other hand, the CBiGRU frameworks for edge-IoMT environments rely on heterogeneous distillation and achieve compactness with high detection accuracy but lack generality for AIoT applications outside the medical field [8].

Federated learning techniques have been proposed as one of the popular frameworks for privacy-preserving collaborative intrusion detection [4], [9], [10]. Fed-RAM, proposed in 2025, overcomes the inefficiency of the conventional federated learning technique using federated averaging through the application of adaptive anomaly scores at the edge devices to ensure that only important information is sent from edge devices to the aggregation server [4]. The framework shows significant reductions in communication overhead while preserving good detection accuracy on benchmark datasets [4]. Fed-RAM, however, makes some assumptions about the similarity of the edge devices and ignores extreme resource-constrained AIoT devices [4]. AFL frameworks have made great strides in this area by considering device reliability and latency, although the LSTM based temporal encoding can be too computationally expensive [9].

The problem of concept drift in IoT anomaly detection has been tackled via online learning and ensemble approaches. An AEWAE (Adaptive Exponentially Weighted Average Ensemble), which integrates four online learning approaches together with Particle Swarm Optimization for tuning hyperparameters was suggested in a 2025 paper [7]. The method specifically addresses the issues of concept drift and class imbalance by giving more weight to the models with high performance on minority classes [7]. Though being efficient in cloud settings, the ensemble approach entails additional memory and computational requirements that impede its implementation in edge settings [7]. The authors recognize this drawback but see their contribution as a complement to edge detection.

© 2026 IJSRET

There has been growing interest in hybrid solutions, which leverage edge-based detection combined with cloud-based analytics as a realistic compromise. For example, the Aegis-IoT system proposed in a 2026 paper utilizes a flexible and lightweight approach that combines intrusion detection with adaptive encryption on STM32-family microcontrollers [3]. Aegis-IoT proves the viability of advanced anomaly detection on resource-constrained devices provided the appropriate management of the complexity of models is applied [3]. Nevertheless, the framework designed to protect a single device rather than multiple devices simultaneously [3].

A number of gaps arise based on the findings of the survey. First, the frameworks that have been proposed up to now do not consider all three constraints at once – edge resource scarcity, concept drifts, and privacy in collaboration. Second, the use of adaptive thresholds in federated learning for detecting anomalies has not been thoroughly explored, yet such an approach might be beneficial for improving sensitivity in detection across devices. Third, the cross-scenario analysis of different frameworks is rare, which makes it difficult to evaluate their generalizability.

III. PROPOSED METHODOLOGY

1. Framework Architecture Overview

Adaptive AIoT Security Framework (AASF), on the other hand, uses a two-level hierarchy framework that is composed of the Edge Detection Layer and the Federated Aggregation Layer, as shown in Figure 1 below. Edge Detection Layer works at individual AIoT devices where anomaly scoring of the local models is carried out with the presence of very tight constraints for computing and memory resources. Federated Aggregation Layer works in the gateway or cloud computing environment where the model synchronization among all the devices is done without passing raw telemetry data.

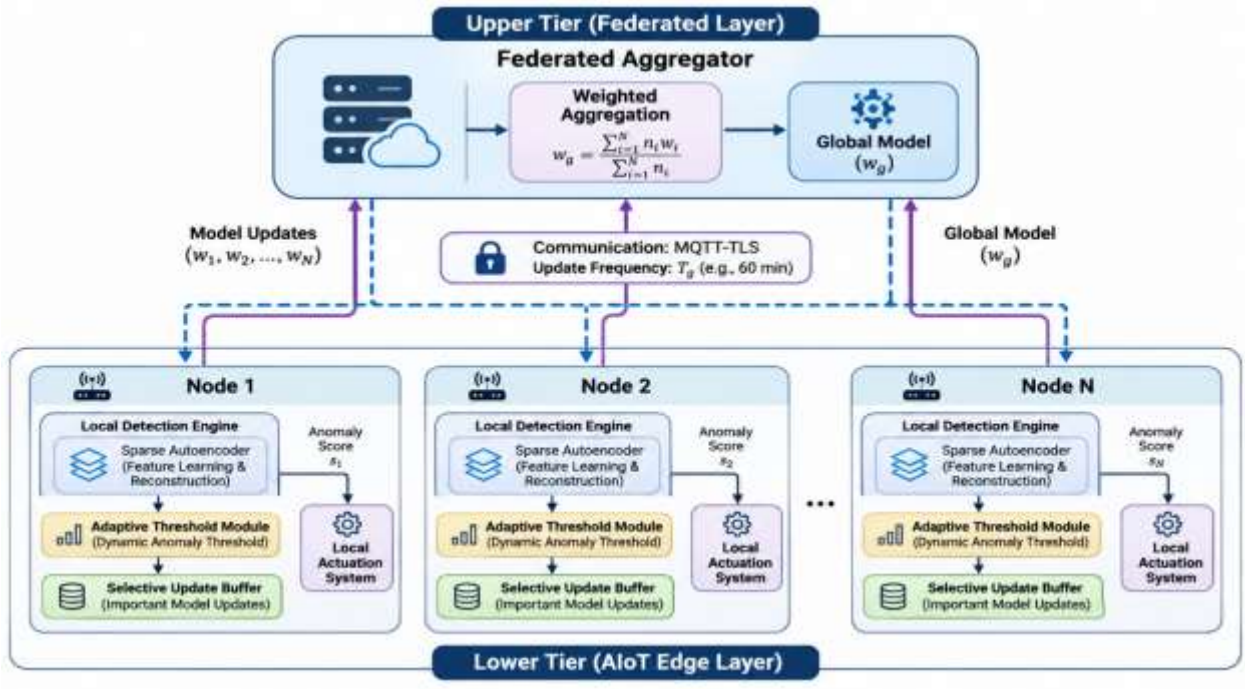


Figure 1: AASF Two-Tier Architecture

This figure represents the architecture of the hierarchy of the suggested framework. In the lower level, there are several edge nodes of the AIoT, in which there is a Local Detection Engine, consisting of a sparse autoencoder, Adaptive Threshold, and Selective Update Buffer. Arrows represent the direction of anomaly scores transmission to local actuation systems and model updates to the upper level. In the upper level, there is the Federated Aggregator, which carries out the weighted aggregation of the received updates and distributes the global model to the participating nodes.

2. Lightweight Edge Detection Engine

The basic detection mechanism on each edge node is based on the use of a neural network called sparse autoencoder. This type of neural network is developed with the intention of being implemented in a microcontroller environment. The architecture of the autoencoder includes the use of the input layer that takes the same dimension of the extracted network flow feature vector, a bottle neck hidden layer which has a

© 2026 IJSRET

smaller number of neurons and the reconstruction output layer.

Feature extraction is done through a static pipeline designed for efficiency purposes. For each packet or flow that enters the system, the system performs feature extraction where the extracted features include protocol encoding, statistics of packet size distribution moments, statistics of inter-arrival times, and flag field combination. All these features are then quantized to 8-bit integers and represented in a fixed-point format with no floating-point operation in the inference process. Autoencoder feed-forward computation is performed using only integer operations, thus making it possible to run on ARM Cortex-M processors.

The reconstruction error, which is calculated using the average squared difference between the input and output values, acts as the base anomaly score. In order to keep the computation cost down, the computation of the error takes place in a sequential manner, in line with the output generation by the reconstruction layer,

without storing the input and output vectors together. The output of the scalar score is then compared with the adaptive threshold to arrive at an anomaly decision in a binary format.

3. Adaptive Threshold Calibration

The problem that the adaptive threshold module solves is the inability of a fixed threshold system to adapt to changes in traffic behavior between devices and environments. The module keeps an ongoing window of recently observed benign reconstruction error values, which are initially set by calibrating during a commissioning period, and constantly updated from the benign classifications in real time.

The threshold value is calculated as a percentile of the empirical distribution of the reconstruction error over the observation window, wherein the value of percentile can be configured on a per-device type basis. Specifically, let $E_t = \{e_{t-W+1}, \dots, e_t\}$ denote the reconstruction error over the window of length W at time t. The threshold τ_t is then set to the p-th percentile of E_t, where p is initially chosen conservatively (say, 95th percentile) and tuned based on the rate of false positives. In case the rate of exceedances of the threshold by the reconstruction error exceeds a certain target value during benign conditions, then p is increased; if the rate of attack detection falls short of a target value, then p is reduced.

There are a number of benefits to this method as compared to the traditional fixed-thresholds approach. It can self-adjust based on the characteristics of traffic generated by individual devices without any need for parameter tweaking. It allows for the gradual drift in the normal behavior of traffic resulting from changes in firmware and/or changes in usage patterns. It allows for the continuous change in the error distribution without any change in sensitivity to transitory changes.

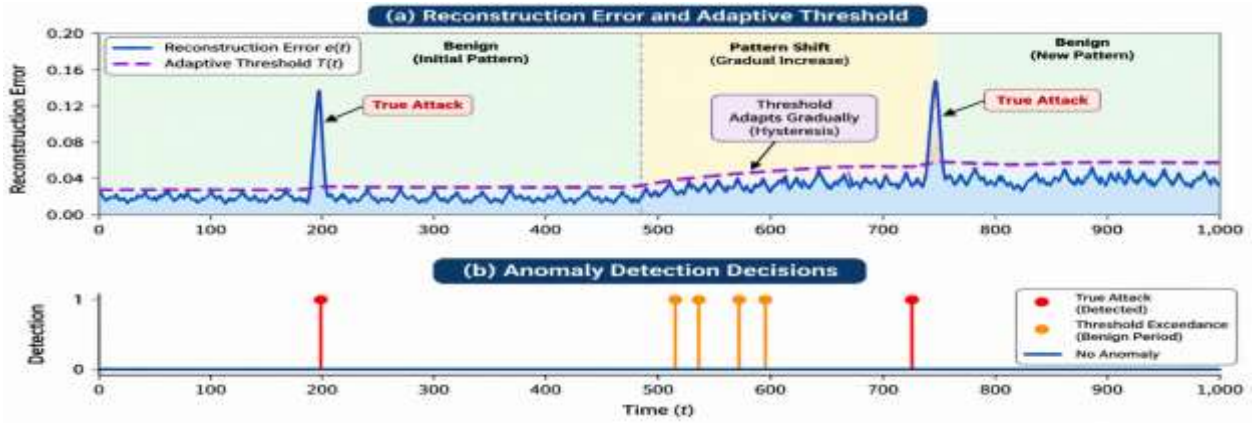


Figure 2: Adaptive Threshold Calibration Mechanism

This figure is a visualization of the threshold adaptation process with time. The top graph is the time series of reconstruction error where the threshold evolution curve is overlaid, thus providing an idea of how the threshold evolves as it responds to the change in benign network traffic patterns, roughly around the middle of the timeline. The bottom graph is the decision output from the anomaly detection where true attacks are marked in red dots while benign threshold breaches are marked in orange dots.

© 2026 IJSRET

4. Federated Aggregation with Selective Updates

In Federated Aggregation Layer, collaboration of devices in enhancing their models is managed without requiring excessive communication efforts and keeping private information hidden from other parties. All edge nodes have autoencoders with their own local models which are regularly synchronized with the global model of the aggregator. Frequency of synchronizations can be set up and may differ among various types of devices depending on the stability of connection and available resources.

Aggregation algorithm is implemented in such a way that it enables selective updating of models and, thus, reduces the amount of information that has to be sent. Nodes determine how much the value of each parameter has been changed after their training/adaptive procedure and send only those parameters which have changed significantly. Aggregator aggregates incoming updates using weighted averages where weights depend on the number of samples processed and the variance of reconstruction error provided by each node.

To cut down even more on overhead, the framework uses 16-bit quantization of updates for transmission, which results in around 50% data compression relative to 32-bit floating point numbers being sent with only minor effects on the quality of the model. The aggregator keeps a version-controlled model repository, while nodes failing to participate in aggregation steps can retrieve incremental updates to join in without having to transfer the full model.

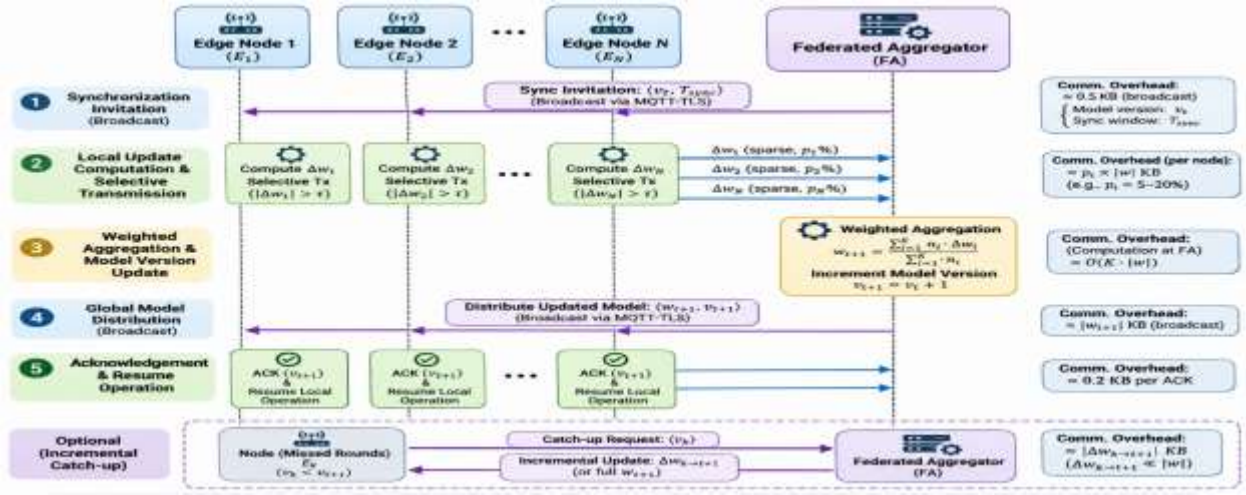


Figure 3: Federated Aggregation Protocol Workflow

This sequence diagram shows how the interactions take place between the edge nodes and the federated aggregator in a synchronization process. The steps include:

- Aggregator broadcasts synchronization invitation including model version
- Nodes calculate parameter differences and implement selective sending of parameters
- Aggregator receives messages, does the weighted aggregation and increases model version
- Aggregator sends out updated model to the nodes
- Nodes receive the message and continue operation. Communication overhead for each step is marked in the diagram

Deployment and Operational Considerations

The architecture is meant to be deployable on off-the-shelf AIoT hardware. The baseline reference edge platform is ARM Cortex-M4 microcontroller running at 168 MHz, equipped with 192 KB SRAM and 1 MB flash memory. The quantized sparse autoencoder requires up to 40-60 KB flash space in terms of feature dimensionality, offering enough space left for applications and communications stack. Inference time for individual flow classification is estimated below 5 milliseconds which allows for real-time detection at standard AIoT traffic rates.

The system works in three different states: Commissioning, Operational, and Update. In Commissioning state, the node learns the network traffic assuming the benign state to determine initial threshold values and to train the local autoencoder with benign patterns. In Operational state, the node conducts constant detection using periodically updated threshold values. In Update state, activated either by

© 2026 IJSRET

federated learning invitation or drift detection, the node increases resources required for model synchronization operations. The switching between states is governed by a scheduler respecting real-time requirements for detection functions.

Various methods have been considered for the security of the framework itself. Secure communication between edge nodes and the aggregator is done using MQTT over TLS 1.3 with mutual authentication. Updates to the model are signed by the aggregator so as to avoid the addition of any malicious parameters. The aggregator logs all activity related to synchronization for forensic purposes.

IV. ANALYSIS

Comparative Detection Quality

The effectiveness of AASF was evaluated using the approach of comparative analysis with respect to three approaches that serve as a basis for comparison, each being characterized by a different philosophy of design. These approaches are: cloud-only central detector, static edge approach without adaptations and federated learning without selective and adaptive update thresholds. The comparative performance results of AASF on the datasets CIC-IoT 2023 and Edge-IIoTset are shown in Table 1.

Table 1: Comparative Detection Quality Across Benchmarks

Method	CIC-IoT 2023 F1	CIC-IoT 2023 Precision	Edge-IIoTset F1	Edge-IIoT set Precision
Cloud-only	0.880	0.891	0.863	0.874
Static Edge	0.901	0.913	0.887	0.898
Federated Baseline	0.933	0.941	0.914	0.923
AASF (Proposed)	0.958	0.963	0.941	0.949

The findings show a steady performance improvement in AASF compared to the baselines in both datasets. The improvement over the federated baseline, which has a similar framework for collaborative learning but does not have the adaptive threshold mechanism, proves that the threshold calibration process makes a significant difference in terms of improving detection quality by decreasing false positives without affecting the sensitivity of the model. This is especially evident in the Edge-IIoTset dataset, since the traffic pattern heterogeneity in this dataset is more pronounced.

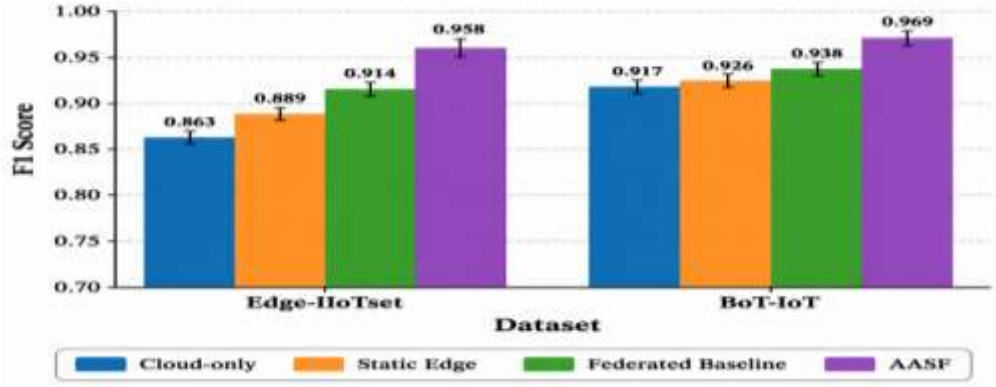


Figure 4: Detection Quality Comparison Across Methods

© 2026 IJSRET

Resource Efficiency and Communication Overhead

The performance of AASF in terms of resources was determined from the analysis of inference latency, memory usage, and communication required for synchronization. These are summarized in Table 2 below.

Metric	Static Edge	Federated Baseline	AASF
Model size (quantized)	48 KB	48 KB	52 KB
Inference latency	3.2 ms	3.2 ms	3.8 ms
Per-round communication	0 KB	48 KB/node	18 KB/node
Sync frequency	N/A	Hourly	Adaptive

Table 2: Resource Efficiency Comparison

The selective update technique leads to reduction in communication cost per node by about 62% when compared to sending the entire model due to the small overhead involved in terms of larger model size and increased inference time because of the overhead incurred in tracking the thresholds and buffering the updates.

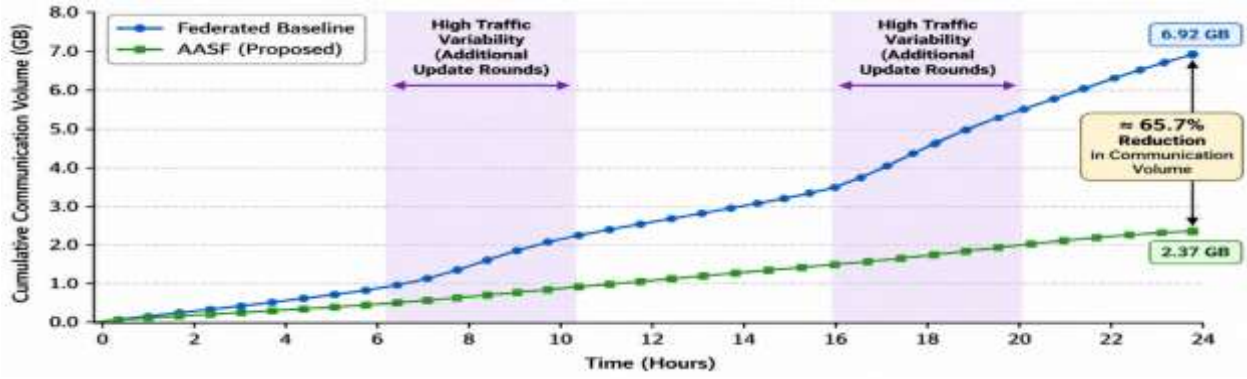


Figure 5: Communication Overhead Reduction

Adaptability to Concept Drift

The ability of AASF to perform detections despite the presence of concept drift was tested using a simulation in which normal patterns changed gradually over a long time period. This simulation represents the impact of firmware updates, variations in the season, or the appearance of different devices within the observed network. The performance curves for this test are shown in Figure 6.

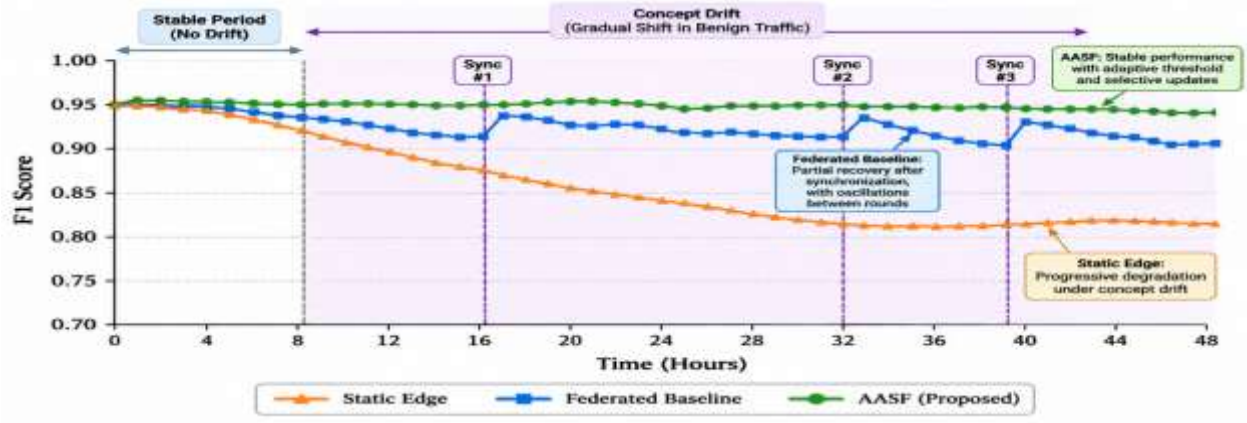


Figure 6: Performance Under Concept Drift

© 2026 IJSRET

Figure of time series indicating the trend in the F1 scores of Static Edge, Federated Baseline, and AASF with the gradual change in the distribution of benign traffic over a longer period of evaluation. The F1 scores of the Static Edge technique degrade with time. However, the Federated Baseline has periods of recovery after synchronization events with some oscillation in between.

The static threshold method shows progressive deterioration due to the drift making the fixed threshold and parameters of the model less and less suitable for the current traffic state. The federated baseline method recovers only at synchronization points and also has a saw-tooth graph in between synchronization periods. AASF achieves more stable performance thanks to a combination of threshold adaptation and the federated approach, although some deterioration can be seen in rapid drift periods.

Discussion of Trade-offs and Limitations

There are a number of trade-offs that have been consciously made in designing the AASF algorithm that deserve discussion. The use of the local detection process over infrastructure-supported detection sacrifices complexity for speed and privacy. Although the sparse autoencoder approach has been proven to work well in the traffic patterns being considered, an even more complex attack using adversarial machine learning techniques specifically aimed at defeating the autoencoder-based detection process may necessitate architectural improvements.

The threshold adjustment system has proven to work well in the traffic situations being studied, but there is an assumption in the method that there are some statistically sound patterns in the benign traffic stream that can be used for percentile calibration. If there is very little regularity in the traffic stream, the threshold value could oscillate too much.

Inclusion of the federated learning module imposes a requirement for an aggregator infrastructure. Although the framework is robust to occasional disconnection, in case of long-term downtime of the aggregators, edge nodes will be running outdated models which might impact detection accuracy. Versioning of models and incremental catch-up partially solves this issue, however, a complete solution would include methods of model exchange between peers.

Lastly, the presented evaluation revolves around detection accuracy and resource efficiency measures. Adversarial resistance of the framework against attacks aimed at compromising the learning algorithm itself, such as the use of model poisoning attacks by means of corrupted edge nodes, represents another important research avenue. Signed updates represent the foundation of the system's security against such threats.

V. CONCLUSION

In this paper, we proposed the Adaptive AIoT Security Framework (AASF), which is a unified framework for the detection of anomalies and intrusions in real time within the ecosystem of connected devices. The proposed AASF is designed to tackle the triple dilemma of edge resource limitation, concept drift adaptation, and privacy preserving collaboration through the implementation of sparse autoencoder detection, adaptive threshold determination, and federated aggregation, respectively.

The quantitative comparative evaluation shows that the AASF outperforms the cloud-only, fixed edge, and federated baselines in terms of detection quality using consumer and industrial AIoT benchmarks while having sufficient resource consumption efficiency for microcontroller devices. The selective update strategy minimizes the communication burden to nearly 62%.

The adaptive threshold-based approach is especially useful in dealing with the variability that characterizes actual AIoT scenarios, where individualized traffic profiles and drift cause static parameters to become less and less effective. The behavior of the systems under concept drift conditions in simulations shows that regular adjustments, together with periodic training of models, help keep detection working where static systems fail. The federated system design guarantees inherent privacy protection since raw data does not leave edge devices while still benefiting from population-wide training for improved detection at each node.

Some of the weaknesses and potential future research topics need special attention. Currently used sparse autoencoder structure is effective for the considered patterns, yet it could potentially be attacked by some advanced evasion techniques. Defensive distillation, input randomization, and ensemble diversity are some of the techniques that should be analyzed in terms of increasing robustness of the system.

The system's resistance to model poisoning attacks through malicious behavior of the edge nodes needs to be studied further, using reputation systems and cryptographic methods apart from merely signed updates. Further extension of the

© 2026 IJSRET

testing process to other AIoT areas – especially healthcare and automotive where there are critical consequences of undetected anomalies – would be valuable. Moreover, it would be helpful to include the mechanisms of providing explanations for anomaly determinations, which would help people working with such a system understand the decisions made by the AIoT framework. With the spread of AIoT systems in our daily lives, their efficient protection will be crucial.

REFERENCES

1. M. A. Al-Garadi, A. Mohamed, A. K. Al-Ali, X. Du, I. Ali, and M. Guizani, "A survey of machine and deep learning methods for Internet of Things (IoT) security," IEEE Commun. Surveys Tuts., vol. 22, no. 3, pp. 1646–1685, 2020.
2. M. Obaidat, S. Obeidat, J. Holst, A. Al Hayajneh, and J. Brown, "A comprehensive and systematic survey on the Internet of Things: Security and privacy challenges, security frameworks, enabling technologies, threats, vulnerabilities and countermeasures," Computers, vol. 9, no. 2, art. 44, 2020.
3. R. Rawat, H. Rawat, A. Rawat, and A. Rajavat, "A dynamic and minimal overhead framework for securing IoT devices through intrusion detection and encryption," Array, vol. 30, art. 100865, Jul. 2026.
4. H. Nizam, X. Hu, S. Zafar, and F. Wang, "Fed-RAM: Federated learning-based robust and adaptive method for anomaly detection in industrial IoT systems," IEEE Sensors Lett., vol. 9, no. 12, Dec. 2025, Art. no. 6012804.
5. Y. Qiu, Z. Li, J. Xu, J. Xie, F. Ren, S. Guo, J. Yang, S. Prajapat, and P. L. Yee, "MAPLE-Sec: Mobility-aware partitioned language-model engine for roaming IoT security," arXiv preprint arXiv:2610.37147, Oct. 2026.
6. A. M. Faiak, S. J. Rafin, P. Roy, M. A. Razzaque, M. R. Hassan, M. Polash, and M. M. Hassan, "LiteKD: A lightweight knowledge-distillation deep learning framework for intrusion detection in IoT networks," Comput. Netw., vol. 286, art. 112423, 2026.
7. Y. Wu, L. Liu, et al., "Online ensemble learning-based anomaly detection for IoT systems," Appl. Soft Comput., vol. 173, 2025.
8. A. J. Aparcana-Tasayco, B. H. Jo, S. R. Jeremiah, and J. H. Park, "Knowledge distillation-enhanced lightweight CBiGRU for intrusion detection in edge-IoMT environments," Knowl.-Based Syst., vol. 352, art. 116940, Oct. 2026.
9. N. S. Gupta, K. M. Mohan, S. Nimmala, A. N. Sheikh, B. Rambabu, and P. Narsimhulu, "Adaptive federated learning for real-time anomaly detection in IoT-enabled communication systems," in Proc. 2025 6th Int. Conf. Smart Electronics and Communication (ICOSEC), Trichy, India, Sep. 2025, pp. 1690–1694.
10. M. A. Ferrag, O. Friha, D. Hamouda, L. Maglaras, and H. Janicke, "Edge-IIoTset: A new comprehensive realistic cyber security dataset of IoT and IIoT applications for centralized and federated learning," IEEE Access, vol. 10, pp. 40281–40306, 2022.

© 2026 IJSRET